HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Design and Engineering 3 (2016) 161–177

www.elsevier.com/locate/jcde

Implementation of persistent identification of topological entities based on macro-parametrics approach

Shahjadi Hisan Farjana[a,*], Soonhung Han[a], Duhwan Mun[b]

[a]*Graduate School of Ocean Systems Engineering, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea*

[b]*Department of Precision Mechanical Engineering, Kyungpook National University, Daegu, Republic of Korea*

Received 28 July 2015; received in revised form 11 January 2016; accepted 11 January 2016
Available online 19 January 2016

Abstract

In history based parametric CAD modeling systems, persistent identification of the topological entities after design modification is mandatory to keep the design intent by recording model creation history and modification history. Persistent identification of geometric and topological entities is necessary in the product design phase as well as in the re-evaluation stage. For the identification, entities should be named first according to the methodology which will be applicable for all the entities unconditionally. After successive feature operations on a part body, topology based persistent identification mechanism generates ambiguity problem that usually stems from topology splitting and topology merging. Solving the ambiguity problem needs a complex method which is a combination of topology and geometry. Topology is used to assign the basic name to the entities. And geometry is used for the ambiguity solving between the entities. In the macro parametrics approach of iCAD lab of KAIST a topology based persistent identification mechanism is applied which will solve the ambiguity problem arising from topology splitting and also in case of topology merging. Here, a method is proposed where no geometry comparison is necessary for topology merging. The present research is focused on the enhancement of the persistent identification schema for the support of ambiguity problem especially of topology splitting problem and topology merging problem. It also focused on basic naming of pattern features.

Keywords: Feature based CAD; Macro-parametrics approach; Persistent identification; TransCAD; Topological element

1. Introduction

Computer aided design (CAD) system can be broadly classified into a solid modeling system (B-rep, CSG, etc.) and feature based parametric modeling systems. In case of B-Rep models (boundary representation model) or CSG models (constructive solid geometry models), features, parameters and attributes could not be modified once they are created. On the other hand, parametric modeling system allows the designer to modify the parameters, features and attributes based on product manufacturing information depending on design requirements. Parametric modelers record the history of the design sequentially so that design intent could be satisfied. This feature favors their popularity among commercial design of product models for the use of collaborative design. Procedural models have the advantage of easy editing of dimensional modification. Collaborative CAD design demands the integration between heterogeneous CAD systems for exchanging CAD data model. For CAD system integration, CAD files are exchanged by using standard file formats (direct translation) or XML based neutral macro file (translation using neutral format mechanism) which retains the design intent.

In case of direct translation design data could be lost with great reduction in file size. Design intent, which means product

*Corresponding author at: Graduate School of Ocean Systems Engineering, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. Tel.: +82 1040719937.
E-mail address: farjana@kaist.ac.kr (S.H. Farjana).
Peer review under responsibility of Society of CAD/CAM Engineers.

http://dx.doi.org/10.1016/j.jcde.2016.01.001

creation and modification history, should not be altered in successful CAD data translation. History based macro-parametric approach solves these issues by using XML format neutral macro file. Both these cases of collaborative design should have conformed between the topological and geometrical entities referenced by a feature between the original model and re-evaluated model. Once a design model is modified, topological entities lost their identity. This problem should be solved for successful CAD data translation by achieving persistent identification of all the referenced entities every time after they are modified. This problem is commonly known as persistent identification problem. The mechanism of persistent identification offers, attaching names to topological entities once they are created based on the operations of creation and retrieving of those topological entities between original and modified model every time after the modification occurs.

Macro-parametric approach is a history based parametric method that enables the designer to exchange the parametric information of CAD models which includes product creation history and modification history. The set of standard modeling commands is defined and used in the format of an XML format neutral macro file.

In history based parametric CAD modeling systems, structure of topological entities should be identified consistently; it means entity naming and entity retrieval should be generic of all the topological entities, independent of all CAD systems and unambiguous to identify persistently. If persistent identification is not generic, it could not be applicable to all types of topological entities associated with different features. If the mechanism is dependent on CAD systems, the mechanism will be different for each type of CAD systems (topology based CAD system/geometry based CAD system). In that case, integration between heterogeneous CAD systems is not possible. And the naming rule should be unambiguous to retrieve once the CAD model is gone through topology splitting and topology merging case, when two or more topological entities have the same basic name. And persistent identification should be done with the minimum data required for CAD model translation; that is feature type, attributes of a feature, parameters of a feature, local coordinates of a feature. Naming can be divided into basic naming and ambiguity solving part. Basic naming involves attaching names with features and topological entities associated with that feature (face, edge, vertices). But in different cases, topological entities could have same basic name; which is defined as ambiguity problem. The ambiguity could arise from topology merging or topology splitting case, while modifying the CAD model. It can also happen during the creation stage of the CAD model. For solving these issues, name matching should be conducted between homogeneous CAD systems (name matching) or heterogeneous CAD systems. Name matching involves two different ways. Local matching between topological entities (1: N comparison) or global matching between topological entities (N: N comparison). Local matching comparison one entity from evaluating model with all the referenced entity is pre-edit model; whereas global matching tries to compare all the topological entities of post-edit model with all the topological entities of the pre-edit model (Fig. 1).

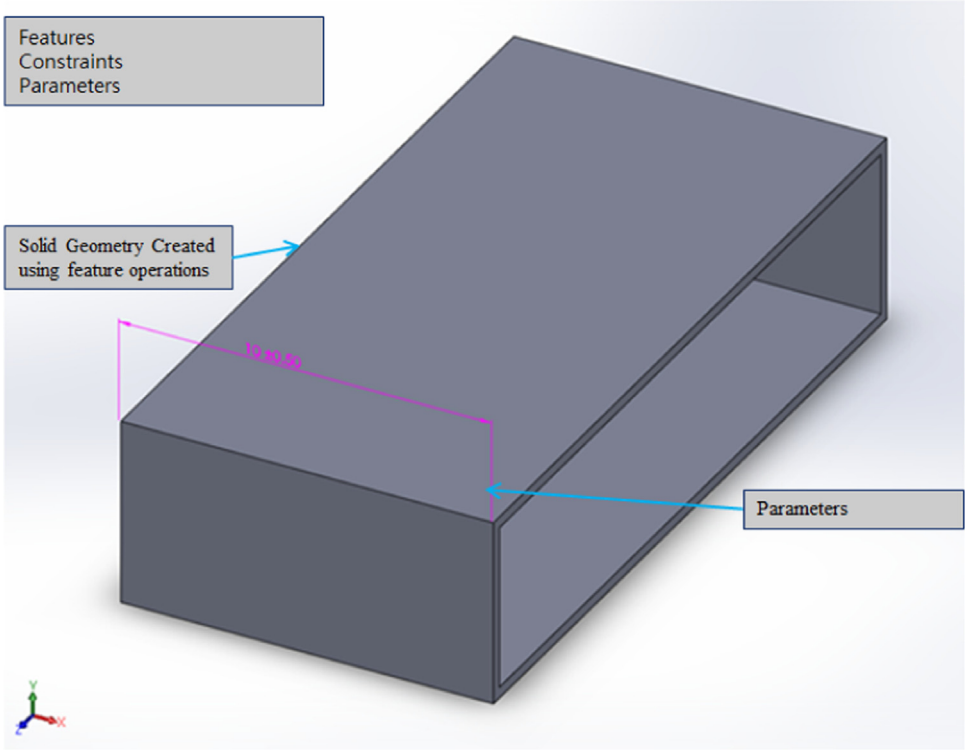

Fig. 1. Components of parametric feature-based solid models.

2. Related works

Kripac [4] proposed a topological system of API (application programming interface) that introduces faceIDGraph describing naming and name matching mechanism when the table is updated every time once the model is modified. Entity matching is based on comparison between one or more faces in two graphs. But his way of differentiating between edges/faces with same basic name is not clear which can lead to unpredictable results [4].

Chen [1,2,3] proposed a topological system of Vertices are matched to define constraints, to construct datum features, to be used by modifying features or to be used as a sub expression in another name. In the dimensioning case, we need to identify the vertex in the three dimensional space. Moreover, only in the case of modifying features we can accept multiple matched vertices with the same name. Edges are named by defining constraints to construct datum features, to be used by chamfers and rounds. In the case of constraints and datum features ambiguous edges can be tolerated if they are collinear line segments with constraint orientation. Faces are matched to determine the limits of feature attachment operations, defining draft operations, identifying a sketch plane or the constraining datum definition. The degree of exactness with which a face is to be matched varies with the operation. This method doesn't consider ambiguity in name matching phase.So as a whole, their limitations are – they tried to name all the entities which are unnecessary & Implementation is difficult [2,3].

Agbodan [5] presented a persistent naming mechanism based on Shell graph [2] which is similar to Kripac's faceID Graph excluding shell and sub shell information. His approach involves hierarchical architecture of the graph which is difficult to implement.

Wu [6] introduced a geometry based ambiguity solving method based on parametric space information (PSI) systems. Parametric space information is calculated from u, v values based on the original names of faces that can generate ambiguity. But Wu did not address any name matching algorithm for that kind of ambiguity problem.

Mun [7,8,10,11,21] proposed topology based basic naming system and name matching system calculated from object space information. Face names are dependent on basic name of the feature where edge and vertex are named depending on that face. But in the case of name matching, Mun considered an ambiguity problem that arises from topology splitting and topology merging case. Naming the ambiguous topological entities that stem from pattern features, that are replica models were out of his scope.

Song [9] introduces a hybrid method (topology + geometry) to persistently identify the entities associated with a feature in case of modeling with an xml based neutral macro file. Geometry based naming never induces an ambiguity problem, but it's difficult to accurately retrieve the reference coordinate information on a feature.

Capoyles [1] described a topology based naming mechanism that involves feature specific information like profile, path; but this method is dependent on features; it's not a generic one. Also edges and vertices are named directly without referencing their adjacent face names.

Raghothama and Shapiro [12] proposed a topological framework for part families depending on its mathematical representation. But this mechanism should be equipped with general solid representation.

3. Comparison with other research

Kripac's proposed algorithm is difficult to implement because details are required for name matching among faces, which were not addressed. Wu's method of parametric space information system based ambiguity solving is not persistent because it is dependent on CAD systems. In the Mun's

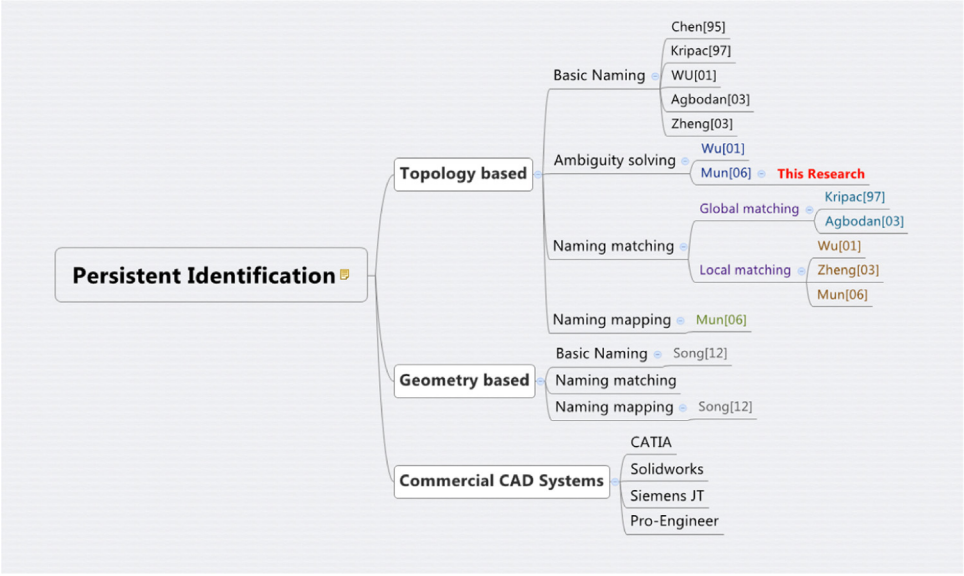

Fig. 2. Macro technology tree of Existing Researches.

method, it is too hard to express the names of features created with multiple loop profile. Also, naming the replica object/ shapes created from pattern feature operation was not addressed; both of them are covered in this study.

Ambiguity problem can be of different types- topology splitting, topology merging, pattern feature, multiple closed loops in one sketch. Several previous researches tried to solve this problem in their methods, but no one is a complete solution to these ambiguity problems. In this paper, we focus on solving the ambiguity problems on the basis of macro-parametric translator, because there are several commercial CAD systems that have their own persistent identification method. But no research or commercial method is sufficient enough to translate the CAD model from one system to CAD model following completely different system (Figs. 2–4). Our focus is a successful CAD model translation for the ease of collaborative design.

4. Persistent identification mechanism

Brep entities that need to be named persistently fall into two categories. One class of entity corresponds to the geometry created explicitly as part of a feature operation. This includes virtually all faces and some edges and vertices. These are associated with a single feature. The second class includes entities that come about through feature collision. Such entities are edges and vertices in which different feature elements intersect. These entities are associated with several features and are transient in the nature, such as after design modification, these entities either modified or deleted.

In history based parametric CAD modeling systems, structure of topological entities should be identified consistently; it means entity naming and entity retrieval should be generic of all the topological entities, independent of all CAD systems and unambiguous to identify persistently.

Fig. 3. Micro Technology Tree of Existing Researches.

Method	Key point	Advantage	Disadvantage	Covered by this research
Kripac et al.	faceIDGraph describing, naming and name matching	Entity matching is based on comparison	Ambiguity problem is not addressed properly. Name table updated every time after modification, so hard to match	Yes
Chen et al.	a topological system of Vertices	Vertices are matched	Doesn't consider ambiguity in name matching phase	N/A
Agbodan et al.	faceIDGraph excluding shell and sub shell information	hierarchical architecture	Difficult to implement	N/A
Wu et al.	Face>Edge>Vertex, PSI based ambiguity solving	Solves the ambiguity problem, also addressed the name matching.	CAD system dependent ambiguity solving method	Yes
Mun et al.	Face>Edge>Vertex, OSI based ambiguity solving	Solving the ambiguity problem and independent of CAD systems	Implementation wise didn't consider the cases when there are multiple closed loops in one sketch,	Yes
Song et al.	Point>Edge>Face	Geometry based method so no ambiguity arises	Not a feature based method, hard to retrieve information about feature operations from topological entity names.	Yes

Fig. 4. Comparison of Existing Researches.

If persistent identification is not generic, it could not be applicable to all types of topological entities associated with different features. If the mechanism is dependent on CAD systems, the mechanism will be different for each type of CAD systems (topology based CAD system/geometry based CAD system). In that case, integration between heterogeneous CAD systems is not possible. And the naming rule should be unambiguous to retrieve once the CAD model is gone through topology splitting and topology merging case, when two or more topological entities have the same basic name.

And persistent identification should be done with the minimum data required for CAD model translation; that is feature type, attributes of a feature, parameters of a feature, local coordinates of a feature.

Naming can be divided into basic naming and ambiguity solving part. Basic naming involves attaching names with features and topological entities associated with that feature (face, edge, vertices). But in different cases, topological entities could have same basic name; which is defined as ambiguity problem. The ambiguity could arise from topology merging or topology splitting case, while modifying the CAD model. It can also happen during the creation stage of the CAD model with pattern feature. Because pattern feature will copy one object in multiple numbers. For solving these issues, name matching should be conducted between homogeneous CAD systems (name matching) or heterogeneous CAD systems. Naming matching involves two different ways. Local matching between topological entities (1: N comparison) or global matching between topological entities (N: N comparison). Local matching comparison of one entity from evaluating model with all the referenced entity is pre-edit model; whereas global matching tries to compare all the topological entities of post-edit model with all the topological entities of the pre-edit model (Fig. 5).

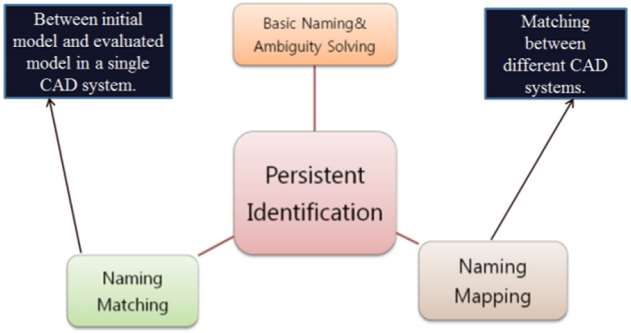

Fig. 5. Mechanism of Persistent Identification.

4.1. Basic naming

Persistent identification can be of two different types in case of basic naming. Topology based or geometry based, depending on the CAD systems. In can also be a hybrid method. CATIA and UG follows topology based basic naming where Solidworks follows a geometry based basic naming. Previous studies show topology based basic naming order can be either of face > edge > vertex or else. In this case, faces are named first and then face names are used to assign names to edges and vertices. Entities can also be subdivided into different categories: invariant entities or contingent entities. In the macro parametric translator of iCAD lab KAIST, faces which are created from feature operations, are named first, then these names are applied to the successive naming of edges and vertices, as edge is a combination of any of two faces and vertex is a combination of at least three faces [19,20] (Fig. 6).

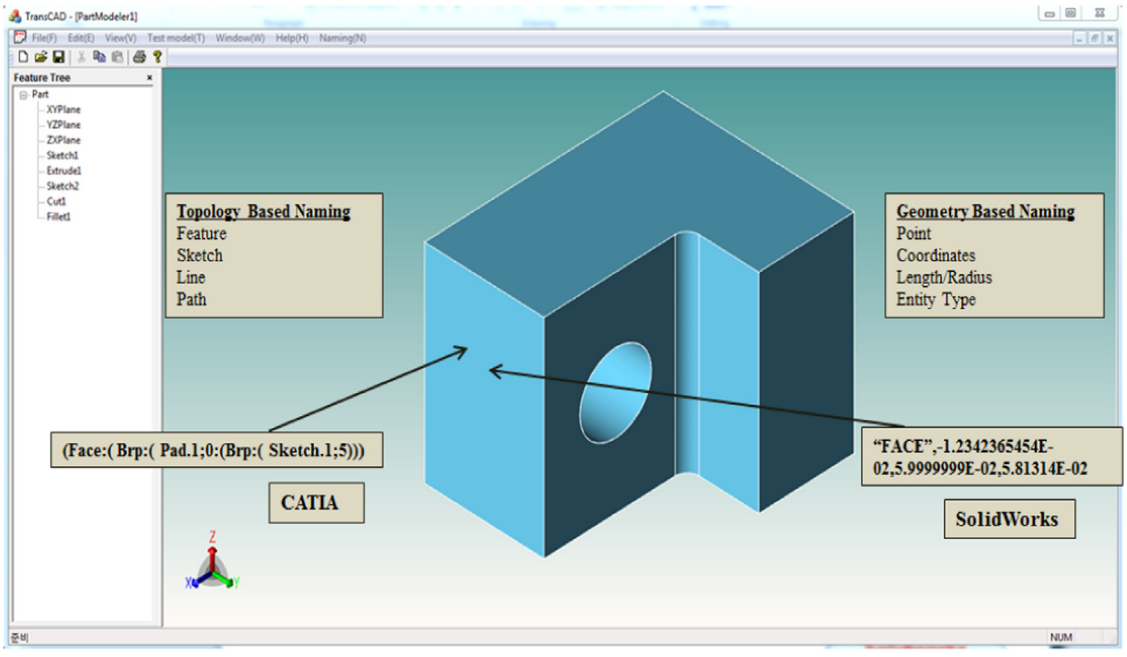

Fig. 6. Basic Naming using topology or geometry.

4.2. Ambiguity solving

Ambiguity occurs in CAD part models when, after feature creation operations (product design phase) or feature editing operation (design review phase), two or more topological entities have the same basic name (topology splitting) or two or more entities whose basic names were different were added into one (topology merging).

Then a different phenomenon arises when basic naming based on topology is not only sufficient to solve this problem. Ambiguous entities can be distinguished using geometry based method, because their geometric dimensions should be some different from the initial one. Here in the diagrams we explained when topology splitting or topology merging occurs.

In topology based naming of iCAD lab KAIST, ambiguity problems are solved by the Object space information system based ambiguity solving method; which enables the geometry comparison of entities and then the new entity name is returned (Fig. 7).

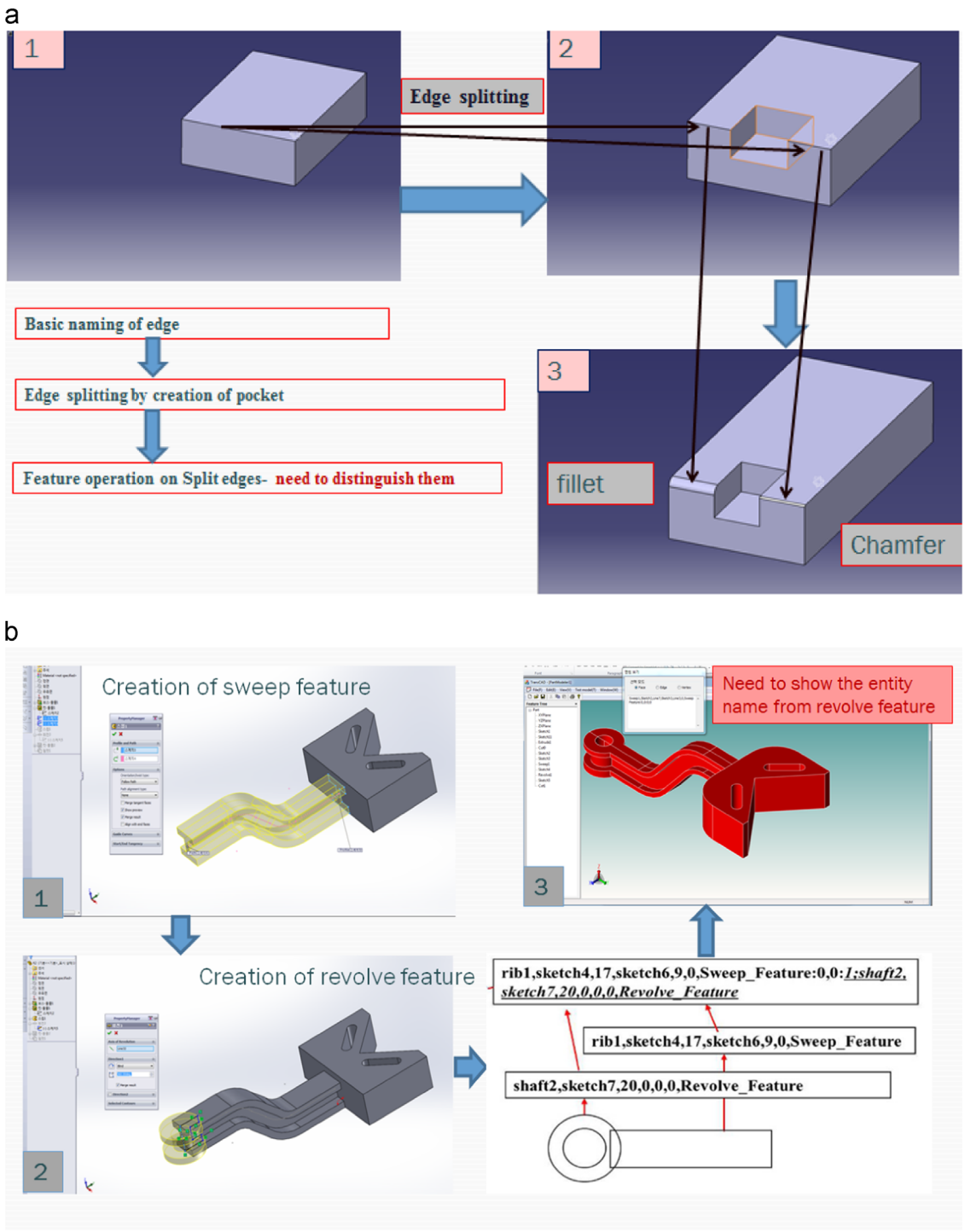

Fig. 7. Ambiguity problem of persistent identification: (a) topology splitting (b) topology merging.

4.3. Naming matching

Naming matching is required between topological entities of CAD models and corresponding topological entities of re-evaluated CAD model. And after matching if there in any modified topological entities than they are given different names from the initial model to identify them persistently. Also, name matching can be of global matching or local matching. Local matching compares identified entity with the other entities in the initial model; whereas global matching compares all the topological entities of the new model with all the topological entities in the old model.

Its N:N matching technique which requires much computation time and skill with greater accuracy. But In case of local matching, it's 1: N which leaves lower accuracy of matching results and fastest computational ability. Different approaches were followed In previous studies.

5. Problems of persistent identification

In K5 model, after cut extrude operation in the revolved and the extruded upper face, inside the macro file of the macro-parametric translator, the name of the cut-extruded entities will be recorded by that feature operation. But if the designer wants to create the linear pattern of that cut-extruded hole, all the pattern instances will copy the same name of the cut-extruded hole; which is not persistent in making the parameter changes.

So to be persistent, pattern objects should contain their specific information; which feature is used for making pattern objects and how many pattern objects are there. Currently, TransCAD names of those pattern objects are wrong and ambiguous.

In K5 model, after cut extrude operation in the revolved lower face, inside the macro file of the macro-parametric translator, the name of the cut-extruded entities will be recorded by that feature operation. But if the designer wants to create the circular pattern of that cut-extruded hole, all the pattern instances will copy the same name of the cut-extruded hole; which is not persistent in making the parameter changes. So, to be persistent pattern objects should contain their specific information; which feature is used for making circular pattern objects and how many pattern objects are there. Currently, TransCAD names of those pattern objects are wrong and ambiguous. So, we need a methodology to solve this kind of problem and for assigning persistent names to them (Figs 10–11).

In K3 model, revolved face is cut into 3 different sections through cut-revolve feature operations. Before the cut revolve operation, the cylindrical face was only one face, which is further divided into 3sections (in the upper face) and the inner faces are also newly created. Without an accurate, persistent identification mechanism, through a CAD model translator, designer will have the loss of design data. Because those newly created faces will copy the basic face name. Persistent identification method demands to identify those new split faces based on their properties (Figs. 8 and 12).

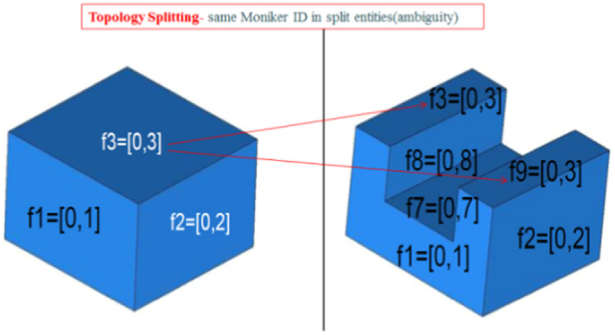

Fig. 8. Persistent identification of Siemens JT of ISO (topology splitting).

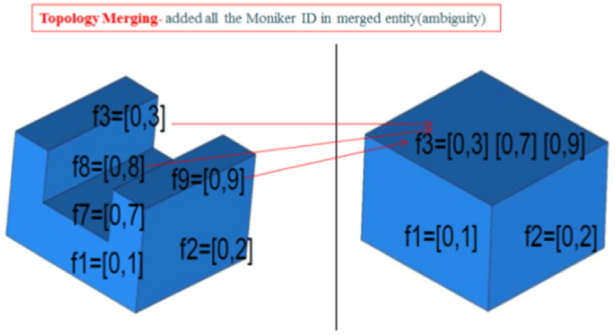

Fig. 9. Persistent identification of Siemens JT of ISO (topology merging).

In K-2 model, the upper face of the bottom part is initially created by protrusion sweep feature; so the entity names are based on sweep feature operations. But after creating revolve feature operation the existing face is modified by adding the circular part. As a whole the face has two types of basic name; one from sweep feature and another from revolving feature. So, the face is a merged face, which should be distinguished from the basic face; but without any topology merging mechanism it's not feasible to identify that merged face (Figs. 9 and 13).

6. Proposed method

In the parametric CAD modeling systems, current persistent identification mechanism attaches name during the product design phase based on their feature operations, sketch ID, path ID, profile ID. This is the initial product creation stage. In the product modification phase, to be specific, as a consequence of successive feature operations on same topological entities for more than one time, ambiguity occurs. These successive feature operation can be pattern feature operations, either rectangular pattern feature or circular pattern feature.

In the product design phase of feature based CAD modeling systems, the topological entities are created and named based on their respective feature operations. Faces are named first, while two adjacent face names are used for naming an edge. Similarly, three adjacent face names are used for attaching names to a vertex. Basic name of the face consists of the feature ID,sketch ID,path ID, profile ID, etc.

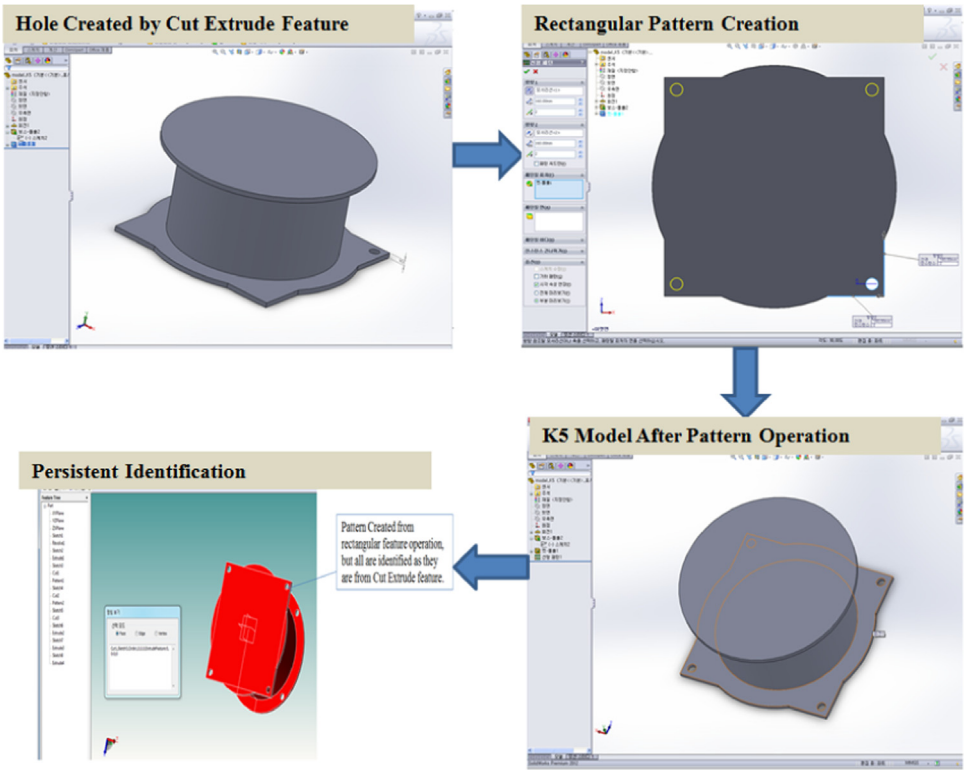

Fig. 10. K-5 model -Lack of identification of feature based entities-Rectangular Pattern.

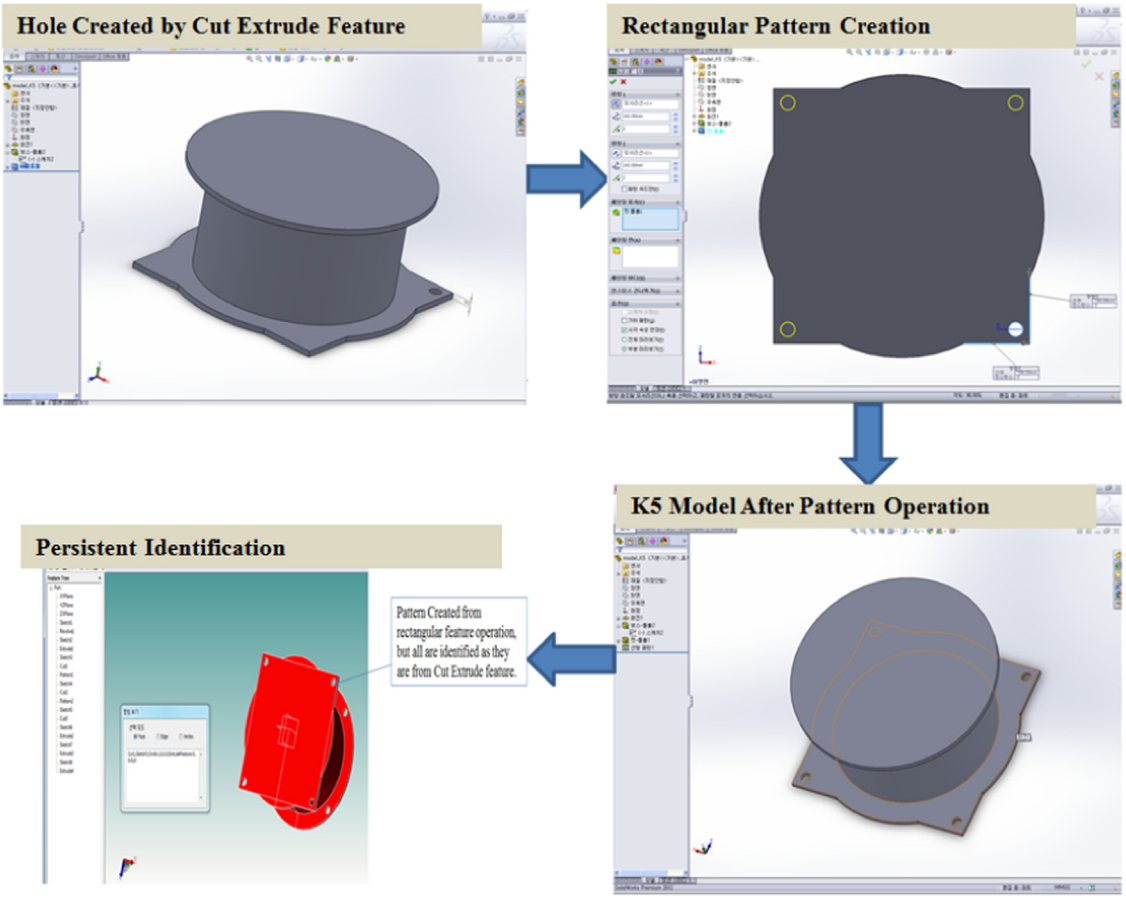

Fig. 11. K-5 model-Lack of identification of feature based entities-Circular Pattern.

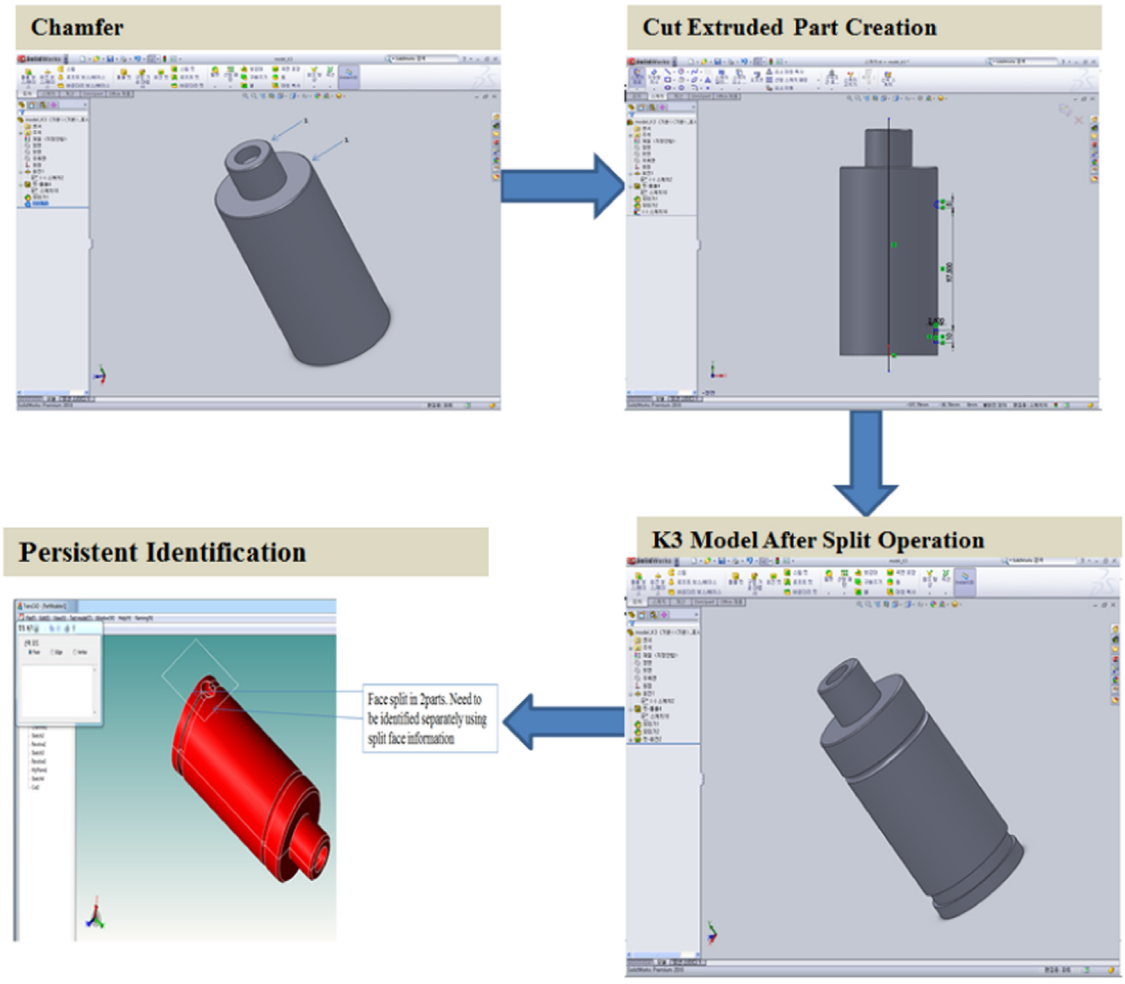

Fig. 12. K-3 model-Lack of identification of split entities.

All these are done in the product design phase. Once a design is modified (a design can be modified by successive feature operation, splitting down the product, attaching new part etc.), the geometry of the topological entities will also modify. As a consequence the modified entities will have new geometric dimensions, but same entity name as before. It's ambiguous, because the modified entity should have a different name from the basic name which will provide the entity modification information. So, we need a specific entity naming mechanism for ambiguous topological entities to persistently identify them after the design modification; which would be able to distinguish them from each other.

Our research focus is an ambiguous entity naming mechanism which is modified during the product design evaluation phase. The distinguished naming mechanism is necessary to find out the difference between design entity and modified entity.

From the process flow diagram of our ambiguity solving mechanism, the persistent identification mechanism first compares the old set of entities with the new set of entities for a matching name. If name is matching then their geometric dimensions should also be matched. If name is matching but geometric dimensions are different one from another, then we have to analyze those entities for ambiguity problem.If after design modification, entities are split into new entities which are created with the same basic name, it's a topology splitting problem and the naming would be based on split face information.

SFI= Order of entity: total number of entities.

Where, order of entity= the higher the geometry, the faster the name will be.

Total number of entity= total number of split entity.

If after design modification, entities are merged into new entity which are created with the different basic name, it's a topology merging problem and the naming would be based on merged face information.

MFI= total number of entities.

Total number of entity= total number of merged entity.

Entity name= basic name1: total number of entity: basic name2.

If after design modification, new entities are created with old basic name but there is no design modification in old entities, then we have to identify whether this problem stems from naming of pattern feature objects.The basic name of the entity should be updated based on feature operation informations (Figs. 14–15).

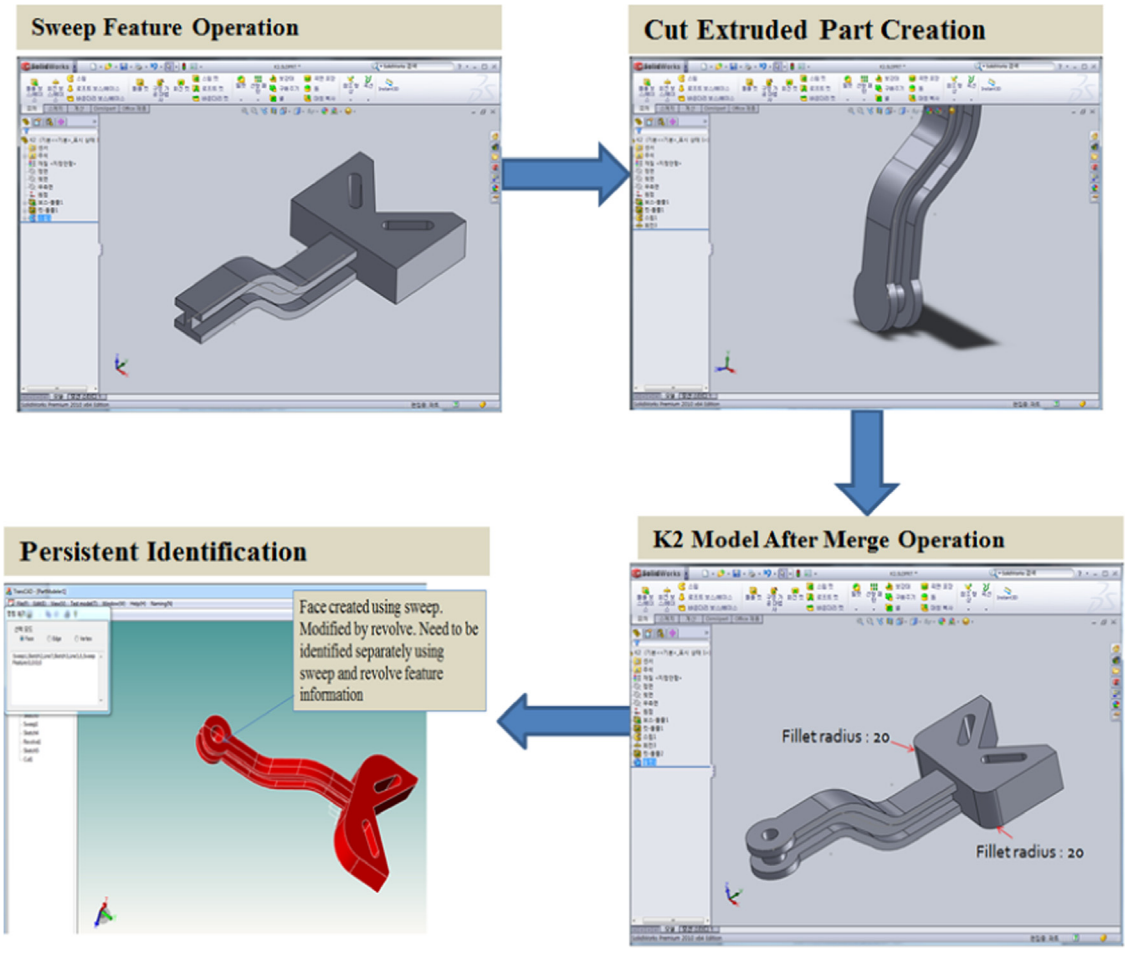

Fig. 13. K-2 Model-Lack of identification of merged entities.

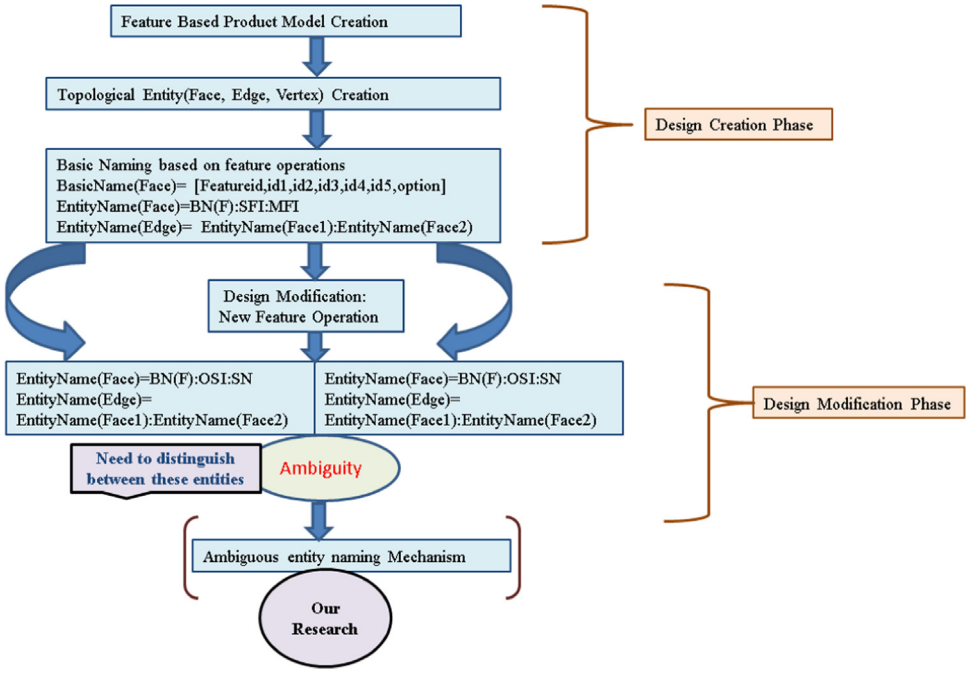

Fig. 14. Process flow diagram for CAD model design and persistent identification.

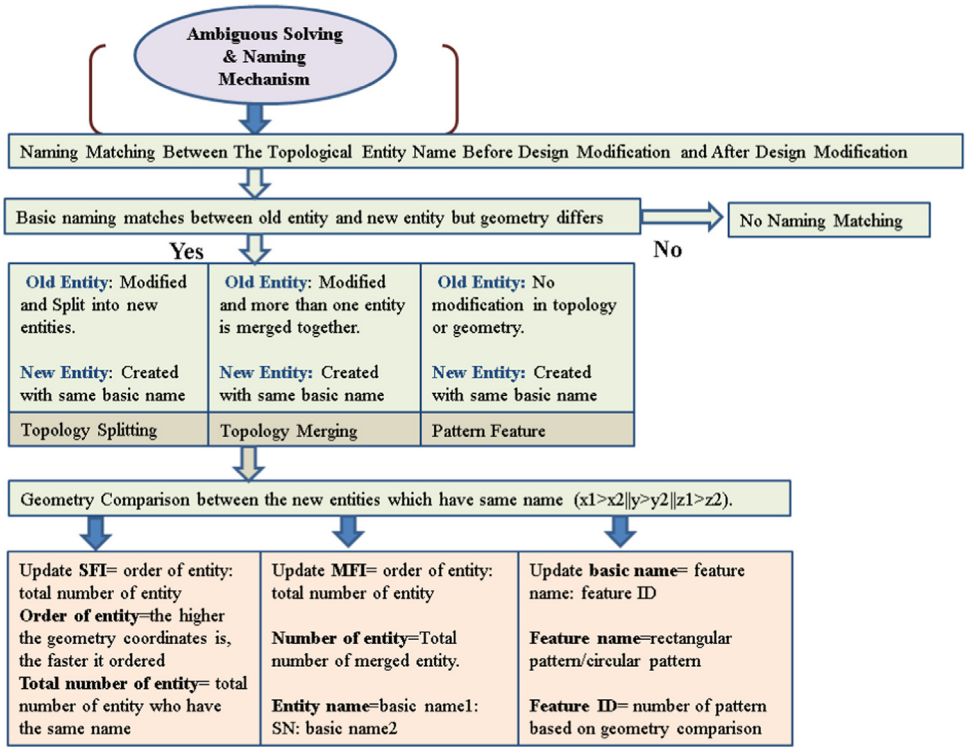

Fig. 15. Process flow diagram for ambiguity solving mechanism for different problems.

7. Implementation environment

TransCAD was developed as a platform between the CAD model translators and the XML format neutral macro file. The figure shows the relation between the translator, the XML macro file, and TransCAD. In this architecture, common modules, such as the geometric modeling kernel and the XML macro parser, are transferred from every CAD translator to TransCAD. As a result, the CAD model translators only have to map the design intentions from TransCAD to the design intent of the receiving CAD system. TransCAD can generate an explicit model from a procedural model, so that each translator does not need to create an explicit model from a procedural model. Different mechanisms are already implemented in the TransCAD. The modification according to the change of the neutral modeling commands set does not propagate to the translators.

TransCAD provides an interacting medium between the translators and TransCAD using the Automation APIs. This Automation APIs, developed by Microsoft, is a COM-based technology which allows developers and designers to create custom applications to automate design tasks (Fig. 16).

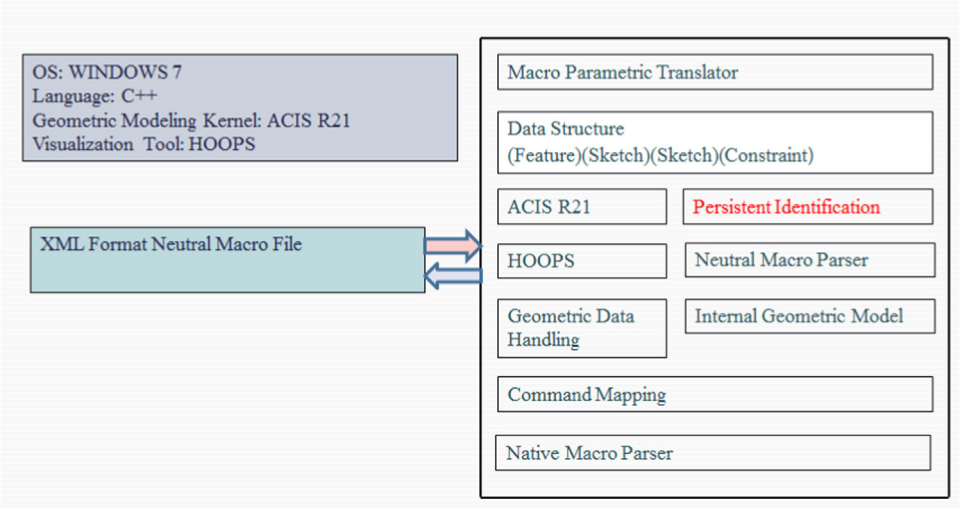

Fig. 16. TransCAD Implementation Environment.

8. Solution algorithm

Based on the feature editing operation implementation, patterned feature based face names are retrieved by comparing the face list before creating pattern objects and also another face list after creating pattern objects. Using ACIS R21 libraries and ACIS API, both of those faces are compared to identify which entities are created by pattern features.

After that identified faces will have a geometry comparison in between them to retrieve the instance number of them. And at last the identified faces will have new names based on pattern feature operations.

According to our method, the basic name of faces of a pattern based features will be:

FaceName= [Feature id, id1, id2, id3, id4, id5, Option]: OSI: SN

=[FeatID, FeatIDp, IDelement, FeatIDpath, IDtrajectory, EntityNum, Option]: SFI: MFI.

Where, SFI=split face information.

MFI=merge face information.

EdgeName= FaceName1#FaceName2

VertexName=FaceName1#FaceName2#FaceName3

For example, after the cut extrude operation the pattern faceName can be like:

FaceName= [Cut2, Sketch4, Circle1, 0, 0, 1, Pattern Feature: 0, 0:0, 0]

EdgeName=[Cut2,Sketch4,Circle1,0,0,1,PatternFeature:0,0:0,0]# [Cut2, Sketch4, Circle1,0,0,0, ExtrudeFeature: 0,0:0,0]

For identification of topology splitting operation, split attributes won't lose their properties. So, the first step is to identify which are those entities. Then, if the entities are split they would have same basic name and different secondary name. Entities with the same basic names are compared to find out the geometry change of them by feature operation through they are split. If the entities are split, definitely they will differ from each other in geometry. A bounding box is made using ACIS API to compare their geometry and get the bigger one. Consecutively the faces will have their secondary names based on their order of splitting operation and number of split face.

According to our method, the basic name of faces of features will be:

FaceName= [Feature id, id1, id2, id3, id4, id5, Option]: OSI: SN

=[FeatID, FeatIDp, IDelement, FeatIDpath, IDtrajectory, InsNum]: SFI: MFI

Where, SFI=split face information= current_number: total_number.

MFI=merge face information.

EdgeName= FaceName1#FaceName2

VertexName= FaceName1#FaceName2#FaceName3.

If the basic faceName is

=[Revolve1,Sketch1,Line1,0,0,0,RevolveFeature:0,0:0;0]

And after another feature operation, this face is divided into 4sections then the child faces will be identified by their secondary names like:

SN(secondaryfaceName)=[Revolve1,Sketch1,Line1,0,0,0, RevolveFeature: 1;4:0,0]

For identification of topology merging operation, merged attributes lost their properties. So, the first step is to identify which are those entities. Then, if the entities are split they would have same basic name and different secondary name. Entities with the same basic name are compared with the initially created entities to check whether they have any clash between them using ACIS API. If they are merged, then those faces are identified by a combination of basic faceName, secondary faceName and number of faces which are merged.

According to our method, the basic name of faces of features will be:

FaceName= [Feature id, id1, id2, id3, id4, id5, Option]: OSI: SN

=[FeatID, FeatIDp, IDelement, FeatIDpath, IDtrajectory, InsNum]: SFI: MFI.

Where, SFI=split face information= current_number: total_number.

MFI=merge face information=Number of faces to be merged

EdgeName= FaceName1#FaceName2

VertexName= FaceName1#FaceName2#FaceName3.

If the basic faceName is=[Sweep1,Sketch2,Line7,Sketch3, Line3,0,SweepFeature:0,0:0;0]

And the secondary faceName is=[Revolve1,Sketch4,Line2, 0,0,0,RevolveFeature:0,0:0;0]

Then the merged faceName will be

FaceName=[Sweep1,Sketch2,Line7,Sketch3,Line3,0, SweepFeature:0,0:0;0]:0,0:1:[Revolve1,Sketch4,Line2,0,0,0, RevolveFeature]

9. Implementation results

In the K4 model, there are three patterned parts are created after the protrusion extrude operation. And the macro-parametric translator TransCAD can identify them based on their instance numbers like 1, 2, 3.

So the basic face name of the feature is=[Protrusion4, Sketch 5,Line 18,0,0,3:Linear pattern:0,0:0,0:]

And for similar two faces= [Protrusion4,Sketch 5,Line 18,0,0,1:Linear pattern:0,0:0,0:]

=[Protrusion4,Sketch 5,Line 18,0,0,2:Linear pattern:0,0:0,0:] (Fig. 17)

In K5 model, there are three patterned parts are created after the cut extrude operation. And the macro-parametric translator TransCAD can identify them based on their instance numbers like 1, 2, and 3.

So the basic face name of the feature is=[Cut1,Sketch3, Circle1,0,0,1,Linear Pattern:0,0:0,0:]

And for similar two faces=

[Cut1,Sketch3,Circle1,0,0,2,Linear Pattern:0,0:0,0:] (Fig. 18)

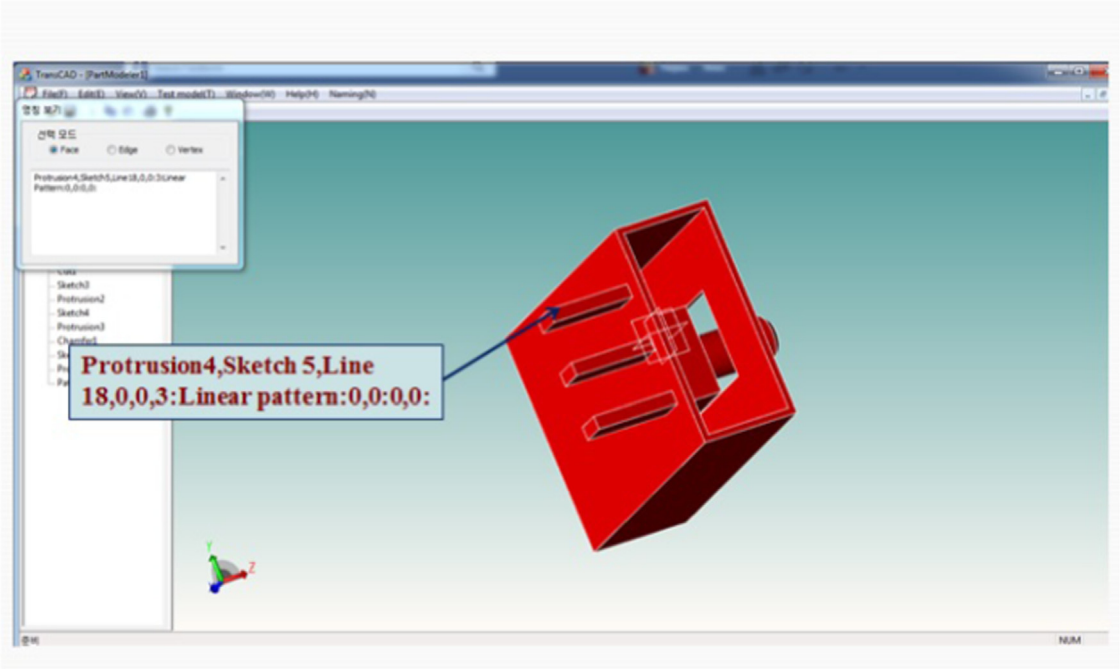

Fig. 17. Implementation status of TransCAD (rectangular pattern) -K4 Model.

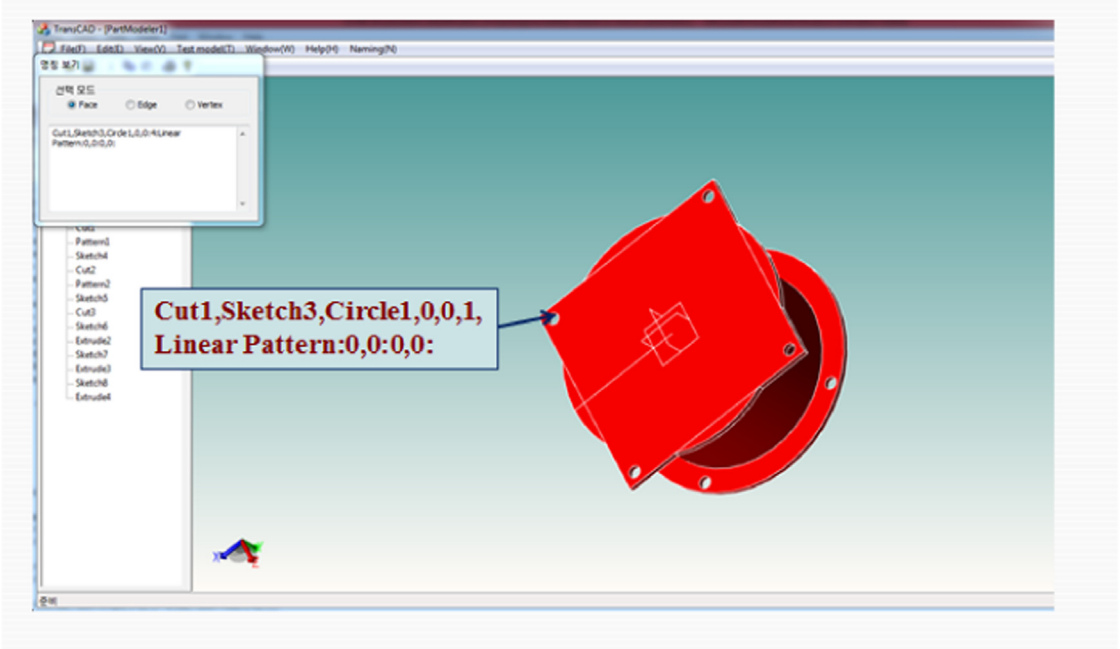

Fig. 18. Implementation status of TransCAD (rectangular pattern)-K5 Model.

=[Cut1,Sketch3,Circle1,0,0,3,Linear Pattern:0,0:0,0:]

In K5 model, there are three patterned parts are created after the cut extrude operation. And the macro-parametric translator TransCAD can identify them based on their instance numbers like 1, 2, and 3.

So the basic face name of the feature is

=[Cut2,Sketch4,Circle1,0,0,3,Circular Pattern:0,0:0,0:]

And for similar two faces=[Cut2,Sketch4,Circle1,0,0,2, Circular Pattern:0,0:0,0:] =[Cut2,Sketch4,Circle1,0,0,1,Circular Pattern:0,0:0,0:] (Fig. 19)

In the K6 model, there are three patterned parts are created after the extrude operation. And the macro-parametric translator TransCAD can identify them based on their instance numbers like 1, 2, and 3.

So the basic face name of the feature is=[Extrude4, Sketch4, Circle2, 0, 0, 3, Circular Pattern: 0, 0:0, 0:]

And for similar two faces= [Extrude4,Sketch4,Circle2,0,0,1, Circular Pattern:0,0:0,0:]

= [Extrude4, Sketch4, Circle2,0,0,2, Circular Pattern:0, 0:0,0:] (Fig. 20)

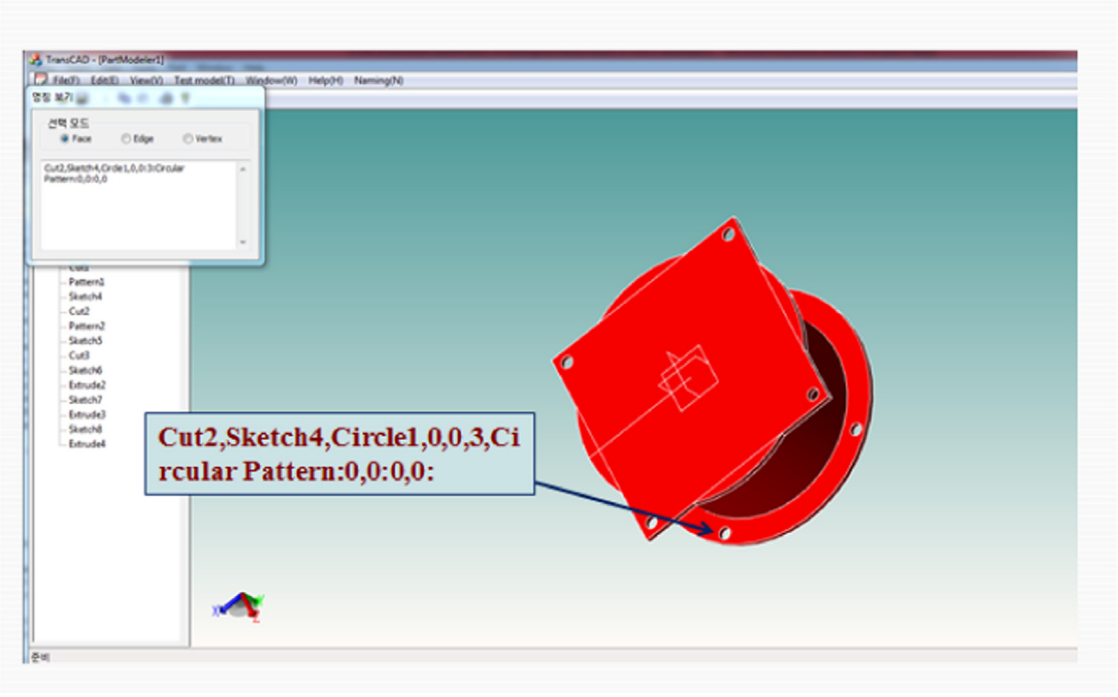

Fig. 19. Implementation status of TransCAD (circular pattern) -K5 Model.

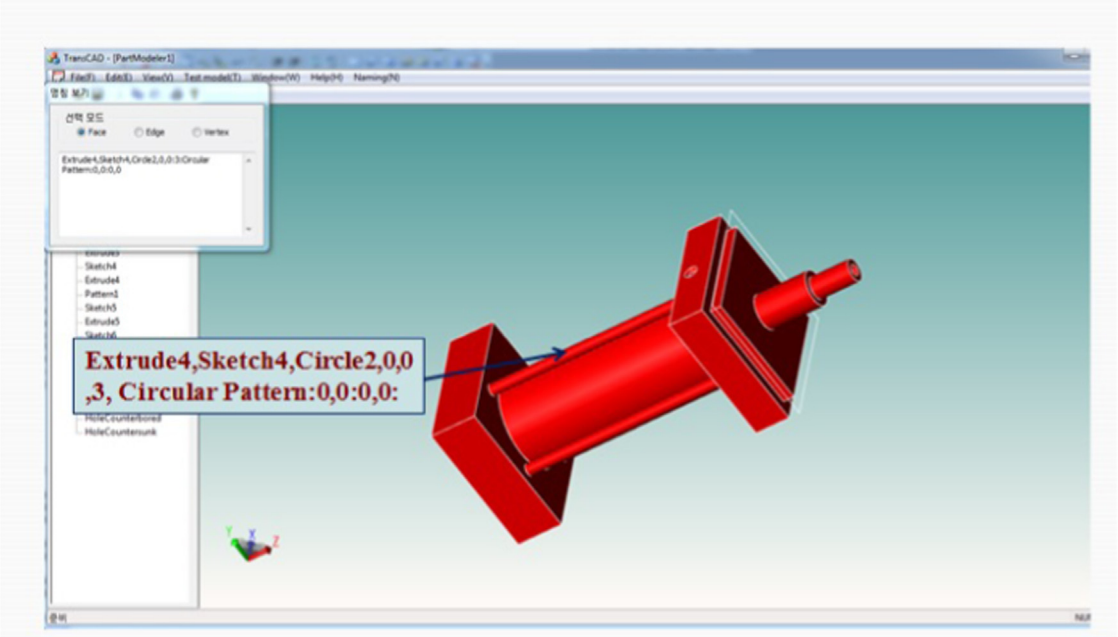

Fig. 20. Implementation status of TransCAD (circular pattern)-K6 Model.

In K3 model, there are three split faces are created after the cut revolve operation. The basic operation was revolve. And the macro-parametric translator TransCAD can identify them based on their instance numbers like 1:4, 2:4, 3:4, and 4:4.

So the basic face name of the feature is=[Revolve1, Sketch1,Line2,0,0,0,RevolveFeature:4,4:0;0]

And for similar two faces= [Revolve1,Sketch1,Line2,0,0,0, RevolveFeature:1,4:0;0]

= [Revolve1,Sketch1,Line2,0,0,0,RevolveFeature:2,4:0;0]

= [Revolve1,Sketch1,Line2,0,0,0,RevolveFeature:3,4:0;0] (Fig. 21)

In K5 model, there are three split faces are created after revolve operation. The basic operation was extrude. And the macro-parametric translator TransCAD can identify them based on their instance numbers like 1:12, 2:12, 3:12, 4:12, etc.

So the basic face name of the feature is=[Extrude1,Sketch2, Line2,0,0,0,Extrude Feature:6,12:0;0]

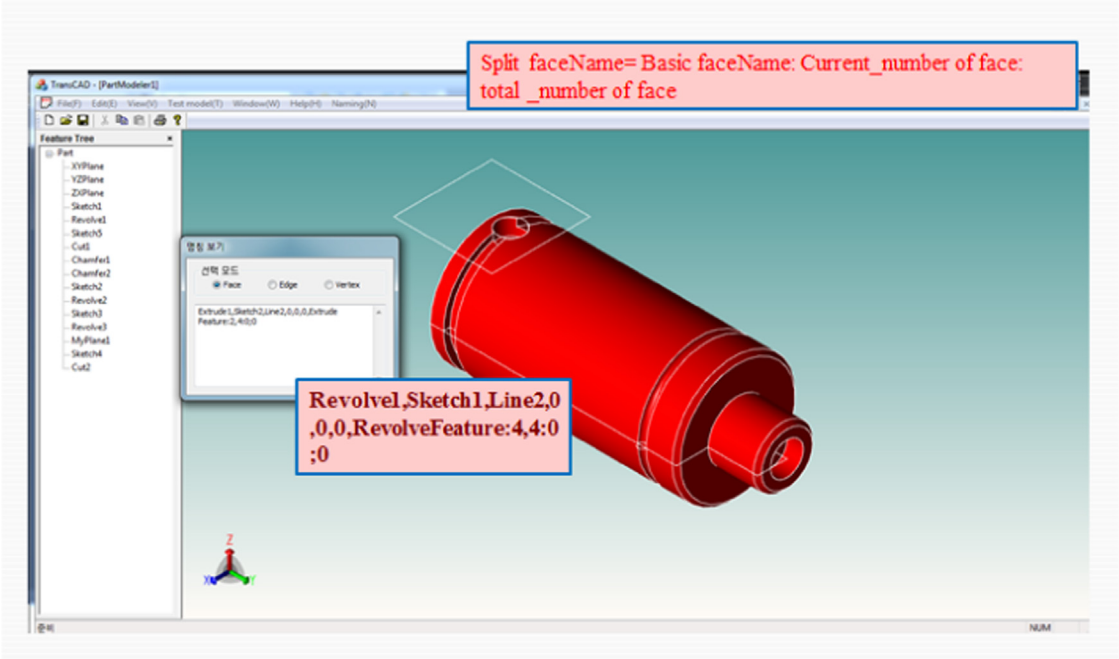

Fig. 21. Implementation status of TransCAD (split case)-K3 Model.

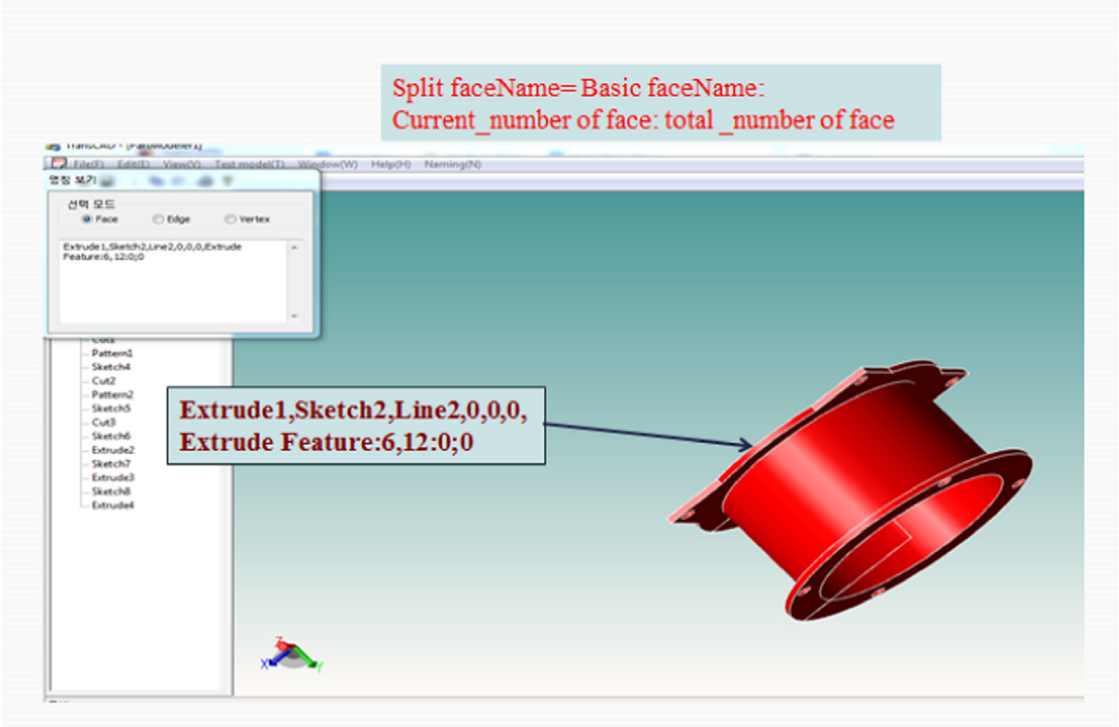

Fig. 22. Implementation status of TransCAD (split case)-K5 Model.

And for similar two faces = [Extrude1,Sketch2,Line2,0,0,0, Extrude Feature:2,12:0;0]

=[Extrude1,Sketch2,Line2,0,0,0,Extrude Feature:1,12:0;0]

=[Extrude1,Sketch2,Line2,0,0,0,ExtrudeFeature:4, 12:0;0] (Fig. 22)

In the K2 model, there are four merged faces are created after revolve operation. The basic operation was a sweep. And the macro-parametric translator TransCAD can identify them based on their both feature operations like:

So the basic face name of the feature is

=[Sweep1,Sketch2,Line7,Sketch3,Line3,0,SweepFeature:0,0:1;Revolve1,Sketch4,Line7,0,0,0,RevolveFeature:] (Fig. 23)

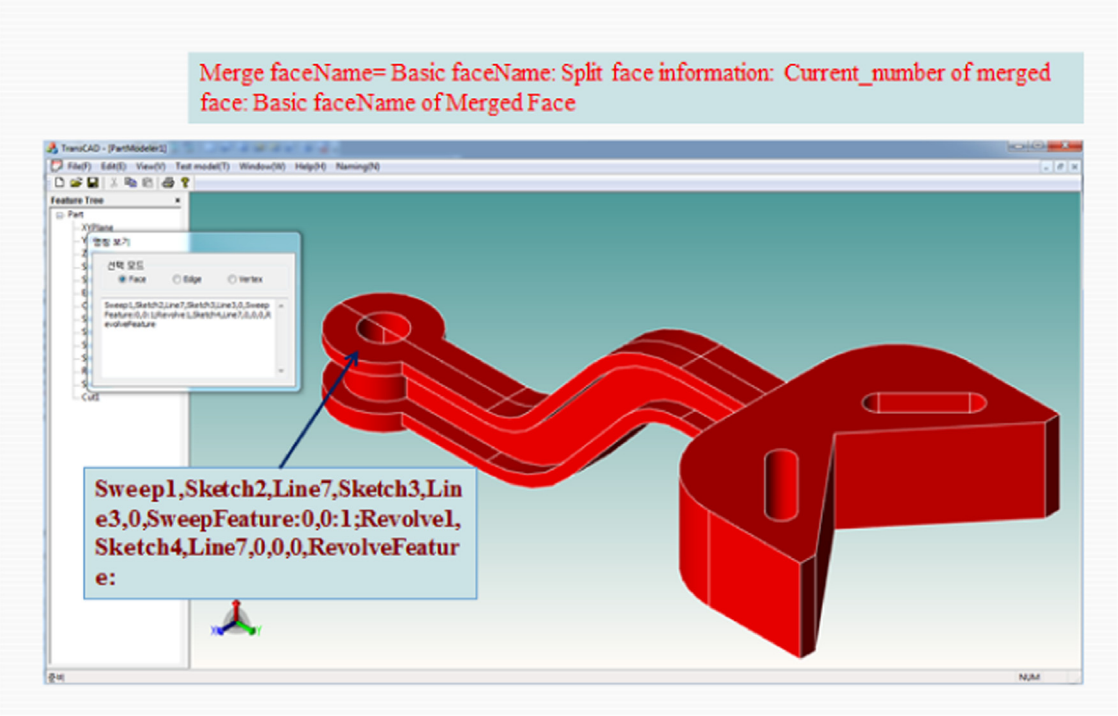

Fig. 23. Implementation status of TransCAD (merge case)-K2 Model.

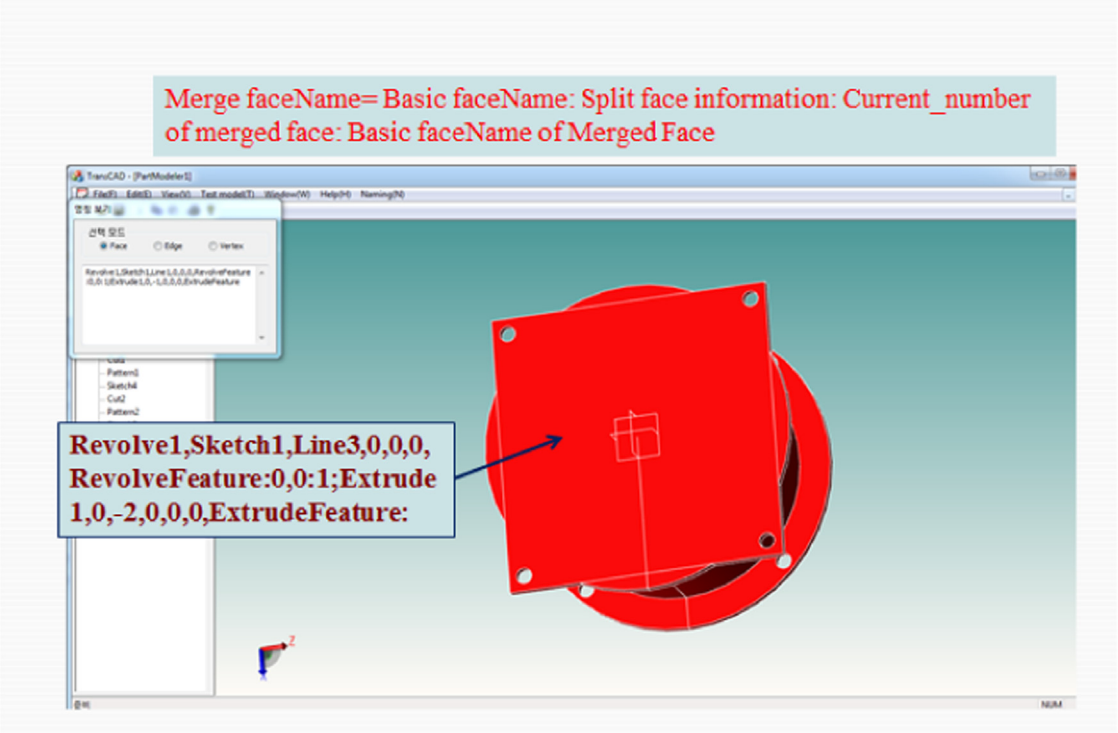

Fig. 24. Implementation status of TransCAD (merge case)-K5 Model.

In K5 model, there are two merged faces are created after the extrude operation. The basic operation was revolve. And the macro-parametric translator TransCAD can identify them based on their both feature operations like:

So the basic face name from feature is=
[Revolve1,Sketch1,Line3,0,0,0,RevolveFeature:0,0:1; Extrude1,0,2,0,0,0,ExtrudeFeature:] (Fig. 24)

10. Conclusions

Our method will solve the persistent identification problem that arises from pattern feature creation for basic identification and also for spilt and merge case while feature editing operations are done. Our method is implemented into history based macro-parametric translator named as TransCAD for converting any CAD model test case which contains patterned object inside itself. Based on the successful translation results our approach will be further extended to other kinds of features which employ feature specific problems during model creation, modification and translation.

Acknowledgment

This research was supported by the Plant Research Program (Project ID: 14IFIP-B091004-01) funded by the Land, Infrastructure and Transport of the Korean Government (Grant number 14IFIP-B091004-01). The authors gratefully acknowledge their support

References

[1] Capoyleas V, Chen X, Hoffmann C. Generic naming in generative, constraint-based design. *Computer- Aided Design* 1996;**28**(1)17–26.
[2] Chen X, Hoffmann C. On editability of feature-based design. *Computer-Aided Design* 1995;**27**(12)905–14.
[3] Chen X, Hoffmann C, Design compilation for feature based and constraint-based CAD, in: Proceedings of the third ACM Symposium on Solid Modeling and Applications (SMA'95), New York, NY, USA: ACM; 1995, pp. 13–19.
[4] Kripac J. *A mechanism for persistently naming topological entities in history-based parametric solid models*. Prague: University of Czech Technical; 1993.
[5] Agbodan D, Marcheix D, Pierra, G, Persistent naming for parametric models, Proceedings of the eighth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media'2000, Czech Republicv: University of West Bohemia, Campus Bory, Plzen-Bory; 2000.
[6] Wu J, Zhang T, Zhang. A face based mechanism for naming, recording and retrieving topological entities. *Computer Aided Design* 2001;**33**(10) 687–98.
[7] Mun D, Han S. Identification of topological entities and naming mapping based on IGM for parametric CAD model exchanges. *Int. J. CAD/CAM* 2005;**5**(1)69–81.
[8] Cheon SU, Mun D, Han S, Kim BC. Name matching method using topology merging and splitting history for exchange of feature-based CAD models. *Journal of Mechanical Science and Technology* 2012;**26** (10)3201–12.
[9] Song I, Han S, Parametric CAD Data Exchange Using Geometry-Based Neutral Macro File, Cooperative Design, Visualization, and Engineering, in: Proceedings of the "7th International Conference, Mallorca, Spain: CDVE 2010 Calvia; September 19–22, 2010 .
[10] Choi G-H, Mun D, Han S. Exchange of CAD part models based on the macro-parametric approach. *International Journal of CAD/CAM* 2002;**2** (1)13–21.
[11] Mun D, Han S, Kim J, Oh Y. A Set of Standard Modeling Commands for the History-Based Parametric Approach. *Computer-Aided Design* 2003;**35**(13)1171–9.
[12] Raghothama S, Shapiro V. Topological framework for part families. *Journal of Computing and Information Science in Engineering* 2002;**2**(4) 246–55.
[19] Solidworks Homepage, ⟨http://www.solidworks.com⟩; 2015.
[20] CATIA Homepage, ⟨www.3ds.com/products/catia⟩; 2015.
[21] Mun D, Han S, Persistent Naming Method Using OSI(Object Space Information) and SN(Secondary Name), Korea CADCAM Conference; 2006.